Research Report

DeepLPI: a novel deep learning-based model for protein-ligand interaction prediction for drug repurposing

Bomin Wei

Princeton International School of Mathematics and Science

Mentors:

Dr. Xiang Gong,

Princeton International School of Mathematics and Science

Prof. Yue Zhang

Division of Epidemiology, the University of Utah, School of Medicine

November 2022

Abstract

The substantial cost of new drug research and development has consistently posed a huge burden for both pharmaceutical companies and patients. In order to lower the expenditure and development failure rate, repurposing existing and approved drugs by identifying interactions between drug molecules and target proteins based on computational methods have gained growing attention. Here, I propose DeepLPI, a novel deep learning-based model that mainly consists of ResNet-based 1-dimensional convolutional neural network (1D CNN) and bi-directional long short-term memory network (biLSTM), to establish an end-to-end framework for protein-ligand interaction prediction. First, the raw drug molecular sequences and target protein sequences are encoded into dense vector representations, which go through two ResNet-based 1D CNN modules to derive features, respectively. The extracted feature vectors are concatenated and further fed into the biLSTM network, followed by the MLP module to finally predict protein-ligand interaction. The well-known BindingDB and Davis datasets are downloaded for training and testing the DeepLPI model. DeepLPI is also applied on a COVID-19 dataset for externally evaluating the prediction ability of DeepLPI. To benchmark the model, DeepLPI is compared with the baseline methods and it is observed that DeepLPI outperformed these methods, suggesting the high accuracy of the DeepLPI towards protein-ligand interaction prediction. The high prediction performance of DeepLPI on the different datasets displayed its high capability of protein-ligand interaction in generalization, demonstrating that DeepLPI has the potential to pinpoint new drug-target interactions and to find better destinations for proven drugs.

Keywords: drug repurposing, deep learning, protein-ligand binding interaction prediction

Introduction

Introducing a new drug to the market has been characterized to be risky, time-consuming, and costly[1][2]. Drug discovery is the first phase of drug research and development (R&D) that starts with identifying targets of an unmet disease, such as proteins, followed by creating and optimizing a promising compound that can interact with the targets efficiently and safely. This step usually involves hundreds and thousands of compounds, yet only about 8% of which drug leads can enter the phase of the *in vitro* and *in vivo* preclinical research [3]. In order to shorten the duration and improve the success rate in the phase of drug discovery, drug repurposing has become a hotspot of new drug research and development over the past few years [1][4], which intends to find an effective cure for a disease from a large amount of existing and approved drugs that were developed for other purposes [1]. For example, prednisone was originally developed for treating inflammatory diseases, but it is likely to be effective against Parkinson's disease as well [5]. In the midst of all the drug repurposing methods, *in silico* computational-based methods to screen pharmaceutical compound libraries and identify drug-target interactions (DTIs) or ligand-protein interactions (LPI) have gained increasing attention and made significant breakthroughs due to the development of high-performance computational architectures and advances in machine learning methods.

Over the last decade, various machine learning-based models have been developed to identify LPI from millions of ligands and proteins. One type of model utilized 3D structures of proteins and drug molecules aiming at capturing interaction details in predictions of the drug-target binding affinity [6], such as Atomnet [7] and SE-OnionNet [8]. However, insufficient 3D protein structure data limited the practicability, generalizability, and accuracy [9, 10, 11]. To exploit the vastly available protein sequencing data, a new model calculates human-selected features and predicts drug-target interactions with conventional machine learning [12, 13]. The disadvantages of these methods are that they require much domain knowledge and possibly lead to a loss of information about raw protein-ligand interactions due to limited features. Deep learning-based models can automatically learn the highly complex and abstract level of features from large-scale datasets without extensive manual creation. Yet the recent development considers only simple encoding of input letter information [14, 15, 16]. Without the contextual information, this type of model may not capture the complex protein features and thus

have limited accuracy and generalizability.

Here, I propose DeepLPI, an innovative deep learning-based model to predict protein-ligand interaction using the simple formats of raw protein 1D sequences and 1D ligands (i.e., drug molecular) SMILES (Simplified Molecular Input Line Entry System) strings as inputs, rather than manual-generated features or complex 3D protein structures. To capture contextual information in the sequence data, I first respectively employ Natural Language Processing-inspired pretrained models of Mol2Vec [17] and ProSE [18] to embed drug SMILES strings and protein FASTA sequences as numeric vectors. These embedded numeric vectors are then fed into two blocks, each of them consisting of two modules termed head convolutional module and ResNet-based convolutional neural network (CNN) module, to encode proteins and drug sequences, respectively. The encoded representations are concatenated into a vector and fed into a bi-directional long short-term memory (biLSTM) layer, followed by three fully connected layers. The BindingDB [19] dataset is used to train the DeepLPI model, adjust the hyperparameters, and independently evaluate the performance of making LPI classification predictions. The model is then applied on Davis [20] dataset to do regression and transformed to predict on a COVID-19 3CL Protease [21, 22] dataset for externally assessing the prediction ability of DeepLPI. To benchmark the model, the regression-adapted version of DeepLPI is compared with the start-of-the-art methods of DeepDTA [15], DeepCDA [16], and DTITR [27] towards protein-ligand interaction on Davis dataset. The prediction performance is quantitively represented in terms of R^2 (higher is better), and Mean Square Error (MSE) (lower is better). The high performance of the DeepLPI towards protein-ligand interaction prediction suggests that the model has the potential to accurately identify protein-ligand interaction and hence, promote the new drug development.

Dataset and data preprocessing

BindingDB [19] dataset is used to train and evaluate the DeepLPI model. Then Davis [20] dataset is used and the COVID-19 3C-like Protease dataset from Diamond Light Source [21, 22] is used for further evaluation and comparison. All datasets are publicly accessible. The BindingDB is a continually updating database that contains $2{,}407{,}235$ experimentally identified binding affinities between $8{,}130$ target proteins and $1{,}036{,}498$small drug molecules up to June 2022. First, the following criteria is applied to compile the dataset for the development of the model: (1) excluding binding interactions with

multichain protein complexes because it is not capable of identifying which chain of the protein interacts with the molecular; (2) retaining binding interactions only represented by K_d value and it means that other measurements in the form of $IC50$ or K_i values are removed; (3) keeping common drug molecules and target proteins occurring in at least three and six interactions in the entire dataset [9], respectively; (4) removing data with invalid K_d values and removing duplicated data entries. For example, some data used">" and "< "in the labeled values to indicate ranges, and therefore they are directly excluded from the subsequent analysis. Additionally, there are some zeros in the values which should not appear based on the definition of binding affinity measurement of K_d. Thus, they are treated as invalid values and are simply removed. As a result, a total of $62{,}825$ interactions with $21{,}148$ drug molecules and $1{,}944$ protein targets are finally used in developing the model. (5) As a binary classification problem in this study, label 1 is used to represent a pair of protein and ligand being active if their corresponding K_d value is less than 100 nM and label 0 is used to represent a pair being inactive if their K_d value is greater or equal to 100 nM since a greater dissociation constant means weaker binding. In this case, 73.5% of data are labeled active, and 26.5% of data are labeled inactive (**Fig. 1a and 1b**).

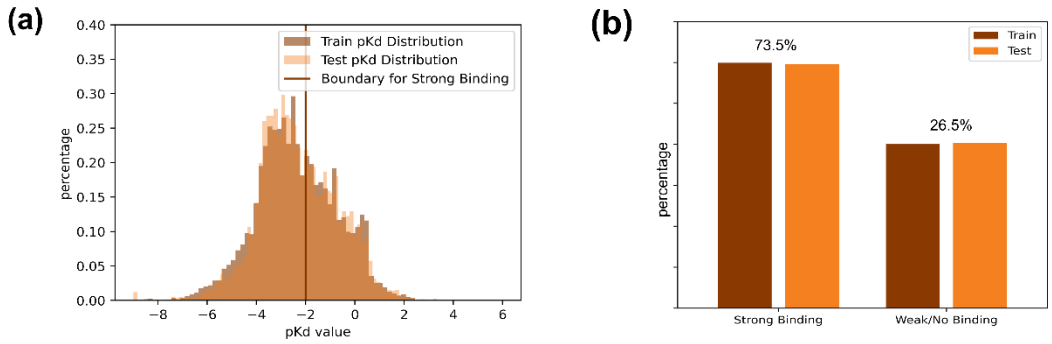

Figure 1. Distribution of BindingDB data used to develop the DeepLPI model. (a) Distribution of the pK_d values and the threshold for determining active/inactive. (b) distribution interaction in binary classes

Four different subsets are built to evaluate the performance of the model (Fig. 2). The test set includes the "Drug Unseen" testing set, which consists of drugs not seen in the training set; the "Protein Unseen" testing set consists of proteins not seen in the training set; the "Both Unseen" testing set consists of drugs and proteins neither seen in the training set; and the "None Unseen" testing set consists both drugs and proteins seen in training, but not the drug-protein pairs. Randomly selected are 15

different ligands and 6 different sequences as test molecules and proteins and used to divide the data sets. 2500 data from the remaining 51,961 data are then selected as the "None Unseen" test set.

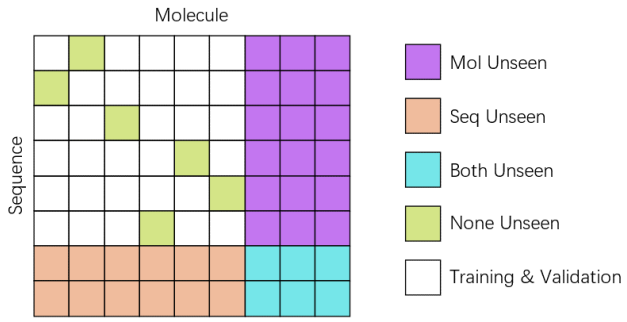

Figure 2. Division of the dataset into training and special testing sets.

The total test set consists of 21% of the entire dataset, and the remaining 79% will be used for training. To optimize hyperparameters, another 1000 data is further allocated from the training set for validation during the training phase, and the rest (i.e., 77% of all data) are used to train the model. The AUROC, accuracy, and confusion matrix are calculated.

The reason for choosing K_d rather than other binding measurements is to better transfer our model to the Davis dataset, which only reports K_d values of the kinase protein family and the relevant inhibitors. the same protocol is used to obtain the class label as mentioned above. The Davis dataset was referenced from the Davis work [20] and downloaded from the URL therein. The dataset is used by several other models, including DeepDTA, DeepCDA, SimBoost, and DTITR, to evaluate the performance. Here it is used as an evaluation dataset which helps to compare the DeepLPI model with other state-of-art drug DTI models. It contained duplicated data entries where the drug-protein pairs are the same, but the binding affinity values are different, potentially due to the experimental conditions. Only one entry is kept in each group of duplicates. After the treatment, there were 24,548 interaction data entries. The data is split into training, validation, and testing sets according to the same method described above.

To find effective drugs for SARS-CoV-2, the model is applied to a COVID-19 dataset where 879 small molecule drugs were tested on the SARS-COV-2 3C-like protease. The experiment measured EC50 results. For classification, label 1 is used to indicate drug-protease active if EC50 is less than 30 nM [23] or 0 representing inactivity. The data is retrieved from a large XChem crystallographic fragment screen against SARS-CoV-2 main protease at high resolution from MIT AiCures. [22] Among those data, 78 are active according to the threshold.

Model Design

Overview of DeepLPI model

The proposed DeepLPI consists of eight modules **(Fig. 3)**, including two embedding modules, two head modules, two ResNet-based CNN modules, one bi-directional LSTM (biLSTM) module, and one multilayer perceptron module (MLP). DeepLPI employs raw molecular SMILES strings and protein sequences as inputs, representing numeric vectors using the pretrained models of Mol2Vec[17] and ProSE[18], respectively. The embedded vectors for the drug SMILES and the protein sequences are then fed into the respective head modules and ResNet-based CNN modules to extract features. The feature vectors for the inputs of drug molecules and protein targets are concatenated, pooled (average-pooling operation), and then encoded by a bi-LSTM layer. Subsequently, the encoded vectors are finally fed into an MLP module, and the final output is activated through a sigmoid function for binary classification to predict active/inactive labels.

Embedding module

To utilize the raw drug molecular SMILES string and protein sequence as inputs to the DeepLPI model, they are first encoded into numeric vector representations using the pre-trained embedding models Mol2Vec[17] and ProSE[18], respectively. Mol2Vec is an unsupervised deep learning-based approach to convert a molecule into a numeric vector representation. Inspired by natural language processing (NLP) techniques, Mol2Vec regards the molecular substructures obtained by the Morgan identifier [23] as **"**words**"** and the compound as **"**sentences**"**, and then encodes them into dense vector representations based on a so-called corpus of compounds. On the other hand, the ProSE is a masked language-based model, using biLSTM networks to capture contextual features in protein FASTA sequence [18], which are represented into numeric vectors that encode protein structural such as structural information and functional properties. It first translates a protein sequence into a list of specific alphabets (as a **"**sentence**")**, which map similar amino acids (as **"**words**"**) into close numbers. Then, the ProSE model encodes the words into numeric vectors. The pre-trained Mol2Vec and ProSE are utilized to obtain vector representations with a fixed length for the drug molecular compound and protein, respectively.

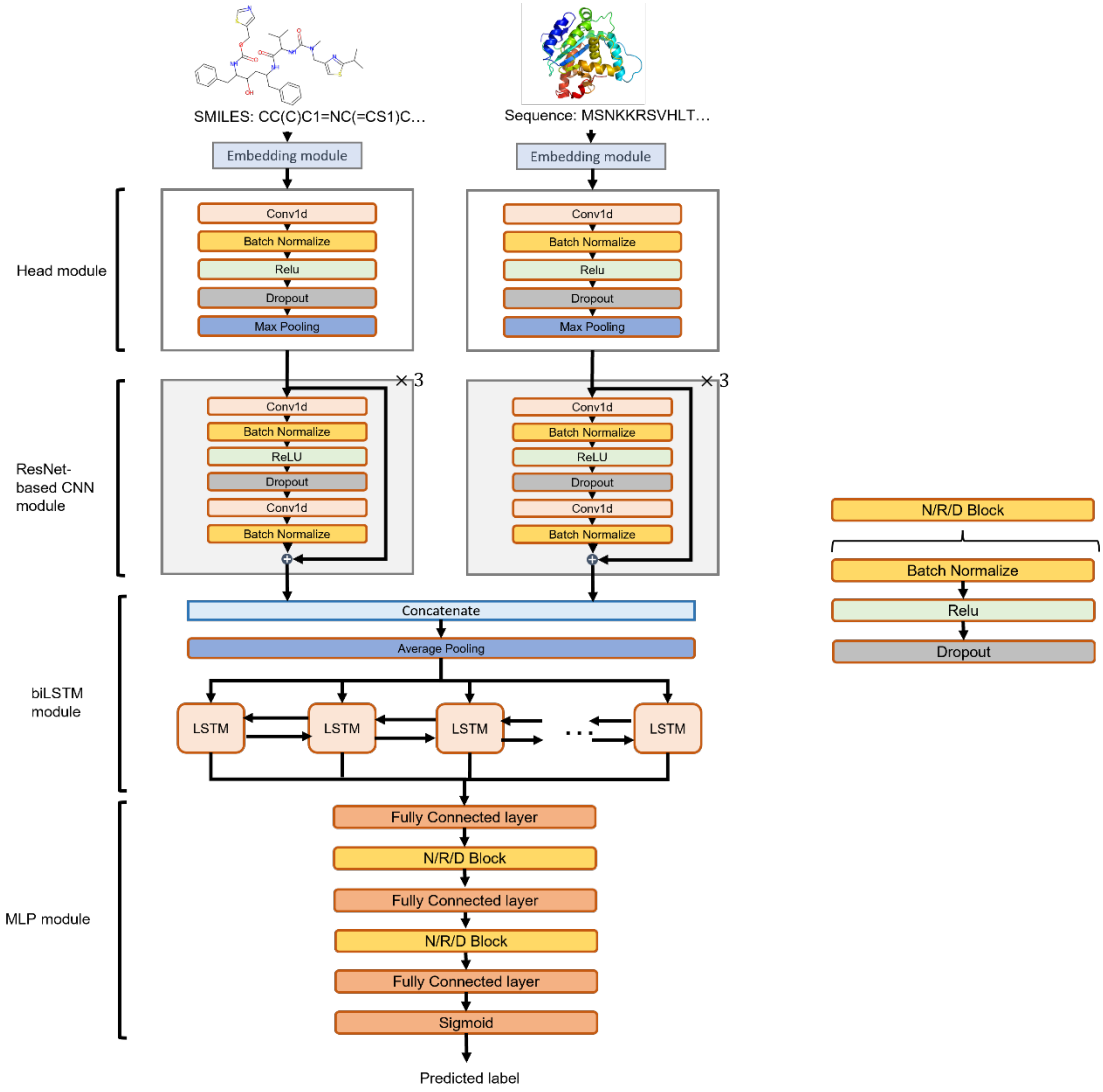

Figure 3. The overview of the DeepLPI model architecture.

Head module and ResNet-based CNN module

After the embedding, the drug molecular SMILES string vector and protein sequence vector are fed into the head modules separately with the same network architecture. The head module contained the following layers: 1D convolutional, batch normalization, nonlinear transformation (with the rectified linear unit, i.e., ReLU activation), dropout, and max-pooling.

Subsequently, two ResNet-based CNN modules are connected to the corresponding head module to encode the input information further. Suppose x is the input into a ResNet-based block. The output of stacked layers is called residual, denoted as $F(x)$, then ResNet-based block output is calculated with

equation $H(x) = F(x) + x$ [24]. Similar to the head module, the two ResNet-based CNN modules had the same network architecture. Specifically, each ResNet-based CNN module consists of three consecutive ResNet-based blocks. Each block comprises two branches, where the right branch is known as a "shortcut connection," and the left branch is known as a "residual network." The "shortcut connection" has two options: for the first of the three blocks, it is a convolution layer, and for the later two blocks, it is exactly the input x. "Residual network" contains several stacked layers, including a 1D convolutional layer, a batch normalization layer, a ReLU layer, a dropout layer, another 1D convolutional layer, and one more batch normalization layer in sequence. The module will output the sum of values that comes out from two branches.

biLSTM module and MLP module

In the biLSTM module, the outputs are concatenate, i.e., the outputs of features extracted by the two ResNet-based CNN modules, followed by an average-pooling layer. After pooling, the concatenated side is directly sent into the units separately. For instance, in the 6165 frameworks, 300-dim molecular embedding and 6165-dim protein embedding translate into 75-dim and 1541-dim separately and then concrete and pool into a 1076-dim matrix. Each unit in the LSTM consists of one of the 1076 inputs, which assumed the matrix as a "time series." The biLSTM, which stands for **bi**directional **l**ong **s**hort-**t**erm memory, can learn long-term dependency from inputs. This network processes the input twice, once from the beginning to the end and once the reverse way, and thus can balance the molecular and protein information. Finally, the outputs on both sides of biLSTM are combined as the output vector.

In the MLP module, flattened is the output vector of the biLSTM and it is fed into three stacks of consecutive fully connected (FC) layers, each followed by batch normalize a Relu activation layer, and a dropout layer. Finally, the output is passed through a sigmoid function for binary classification to predict 1/0 labels. Specially, the model is also adapted to doing regression in order to compare with other methods. In the regression cases, the sigmoid activation function will change from the last to the second consecutive FC layer.

Loss function

When treating the prediction as a classification task that predicts whether the drug and protein have a strong or weak binding, the loss function of n pairs of molecular SMILES strings and protein sequences was given by:

$$\text{Loss} = \underbrace{-\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(\hat{y}_i) + (1-y_i) \cdot \log(1-\hat{y}_i)]}_{\text{BCE loss}} + \underbrace{\alpha\|W\|_2^2}_{\text{L2-norm regularization}} \quad (1)$$

Where $y_i \in \{0,1\}$ is the class label representing whether or not the binding interaction of an input pair of protein and ligand sequences i. $\hat{y}_i$ is the probability of interaction prediction for the input pair i by the model, $\hat{y}_i = sigmoid(x) = \frac{1}{1+e^{-x}}$, x is the output of the MLP module of the model. W is the trainable weight matrix in the model. α is the decay rate, and set to be 0.8 in this study.

In other cases, when adapting the model to the regression task, which predicts the pK_d value without changing it into a 0/1 label, the loss function of n pairs of molecular SMILES strings and protein sequences is also changed to:

$$\text{Loss} = \underbrace{-\frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{y}_i]^2}_{\text{MSE loss}} + \underbrace{\alpha\|W\|_2^2}_{\text{L2-norm regularization}} \quad (2)$$

Parameters Setting

Kaiming Initialization Method is used to initialize DeepLPI network weights [25]. The Adam optimizer [26] is also employed with default parameters of $\beta 1 = 0.9$ and $\beta 2 = 0.999$ as an optimization algorithm to train the model. Furthermore, a batch size of 256 is used and a learning rate 0.001 is used with a decay rate of 0.8 by the ReduceLROnPlateau [25] scheduler. The maximum number of epochs sets to 1000, and the evaluation result on each epoch is recorded on Tensorboard. All settings for the parameters implemented in the DeepLPI model are demonstrated in Table S1. It should be noted that the regressional adaptation version of the model has slightly different parameter sets. It's because the difference between activation and loss functions makes the probability of overfitting different. The parameter values for the pre-trained Mol2Vec and ProSE are set to default. The method yielded vector representations with a fixed length of 300 for the drug molecules and two lengthes of 100 and 6,165 for the target proteins. The 6,165-element representation for protein were tested to outperform the 100-element representation, and thus in the article only the 6,165-element result is

reported. Generally, the hyperparameters of the DeepLPI is manually tune and the model is optimized such as choosing the number of blocks empirically in the ResNet-based module on the Binding DB database, so the Davis dataset's performance can partially reflect the transferability of the model.

Model Evaluation

For the binary prediction, it is desired to calculate the AUROC and Accuracy to evaluate the performance of the model. The confusion matrix is also calculated and shown to further evaluate the performance of the model in specific situations. AUROC refers to the area under the receiver operating characteristic curve, which describes the model's performance in different thresholds by showing the relation between sensitivity and specificity when increasing the threshold,

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

In the above equations, TP represents true positives, FP represents false positives, TN represents true negatives, and FN represents false negatives. Sensitivity indicates the probability of having a correct positive prediction in all cases labelled true. Specificity indicates the probability of having a correct negative prediction in all cases labelled false.

Experiment setup

Model training was done in Aliyun Cloud Computing. The node CPU used Intel(R) Xeon(R) Platinum 8163 (2.50GHz). An Nvidia Tesla T4 GPU is supplied. The model is implemented using the PyTorch library (version 1.8.1). The source code of training and evaluating DeepLPI and the requirements are available on GitHub (https://github.com/David-BominWei/DeepLPI).

Results

Evaluation on BindingDB Dataset

I first report the result of the model trained on the BindingDB dataset. This represents the core performance of the DeepLPI because the hyperparameters are tuned based on this dataset. The model's training stopped at 750 epochs when the learning rate decreased to the minimum of 0.00001. Training after this point would not be able to improve the result significantly and may lead to overfitting.

The model is applied to the test sets, and the AUROC, Accuracy, and Confusion Matrix is then calculated to evaluate the model's performance on the four test sets (Fig. 4a): "Molecule unseen" (Fig. 4b), "Protein unseen" (Fig. 4c), "None Unseen" (Fig. 4d), "Both Unseen" (Fig. 4e). The best accuracy of 0.91 and best AUROC 0.942 are achieved at the "None Unseen" test set, which means the model has an optimistic performance on classification when both the molecule and protein have occurred in the training set. The AUROC results of "Molecule unseen" and "Protein unseen" reach 0.783 and 0.862, respectively, and show that the model has a better performance when only the disease is unknown, which suits the purpose of the model to repurpose existing drugs for an unknown disease. Especially, the confusion matrix shows a very high specificity, which means the model has a very good ability to screen out the inactive drugs which cannot cure the disease. This could help expedite drug discovery in wet lab and clinical tests by reducing the list of candidate drugs needed for experiments.

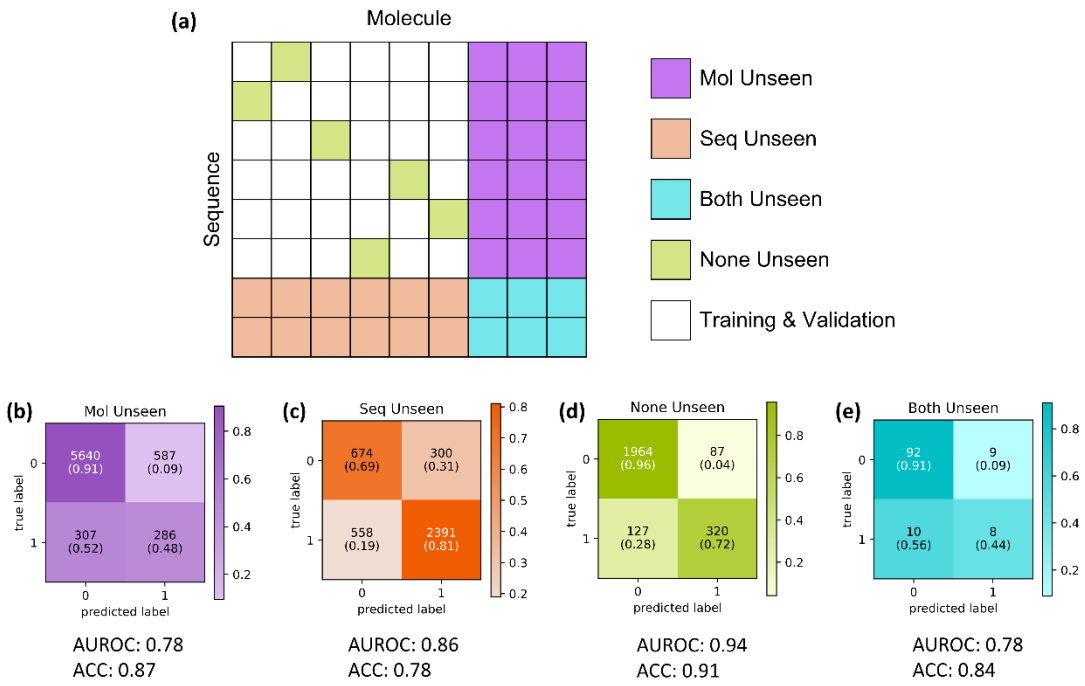

Figure 4. The prediction performance of the DeepLPI model on BindingDB dataset. (a) The color coding of the testing set division scheme. Accuracy, AUROC and confusion matrix are shown on testing set of (b) "Molecule Unseen" (c) "Protein Unseen" (d) "None Unseen" (e) "Both Unseen"

In order to demonstrate the performance of the model, the performance of the model was compared with the recently published DeepCDA [16] model and the popular baseline model DeepDTA [15] on classification tasks. DTITR model is not included in this comparison because its encoding module cannot process phosphorus. it was noticed that the predicted labels from the DeepCDA model on all "unseen" test sets were all zero, indicated by the sensitivity (0.0) and specificity (1.0) (**Table 1**), and cannot make meaningful predictions. It is also found that the DeepLPI model had relatively high generalizability in making predictions for "unseen" test sets. The prediction performance on the "None Unseen" test set, where both molecules and proteins were individually used in training, were summarized in the **Table 1**. The results demonstrated that the DeepLPI scored higher AUROC by 0.060 and 0.053 than the DeepCDA and DeepDTA, respectively. It showed that the DeepLPI can predict in the BindingDB dataset with higher ability of clasificiation.

`

Table 1. Comparing Performance of DeepLPI, DeepCDA and DeepDTA on the "None Unseen" test set from the BindingDB dataset.

Model	AUROC	Sensitivity	Specificity
DeepLPI	**0.942**	0.483	**0.910**
DeepCDA	0.882	0.792	0.804
DeepDTA	0.889	0.772	0.862
Unseen Testsets Combined			
This work	**0.790**	**0.684**	0.773
DeepCDA	0.448	0.000	**1.0**

Transferred Evaluation on Davis dataset

We further applied the trained DeepLPI above in the Davis dataset and compared its performance in both classification and regression tasks with the reported results from the DeepCDA and DTITR. The classification training on the Davis dataset was stopped after 850 epochs. The DeepLPI model obtained an AUROC of 0.923, an accuracy of 0.851, a sensitivity of 0.93, and a specificity of 0.73. In the Table 2, the performance metrics of the model were compared with DeepCDA, DeepDTA, and DTITR on the testing set from the Davis dataset. The testing set is randomly split from the Davis dataset, which includes 16% of the data.

Table 2. Comparing Performance of DeepLPI, DeepCDA and DeepDTA on the independent testing set from the Davis dataset

Model	AUROC	Sensitivity	Specificity
DeepLPI	0.923	**0.930**	0.730
DeepCDA	0.912	0.766	**0.896**
DeepDTA	0.909	0.865	0.795
DTITR	**0.932**	**---**	---

The DeepLPI scored higher AUROC values by 0.011 and 0.014 than the DeepCDA and DeepDTA, respectively, and slightly lower than DTITR's score of 0.932. This result showed the transferability of the DeepLPI model. Given that all models' AUROC is above 0.9, the differences in AUROC performance are not big enough to indicate a superior model. Since the hyperparameters in the DeepLPI were tuned in the BindingDB dataset and then directly applied to the Davis dataset, the DeepLPI performance in the Davis dataset should be further improved if the DeepLPI model was trained in the

Davis dataset like both the DeepDTA and DTITR did.

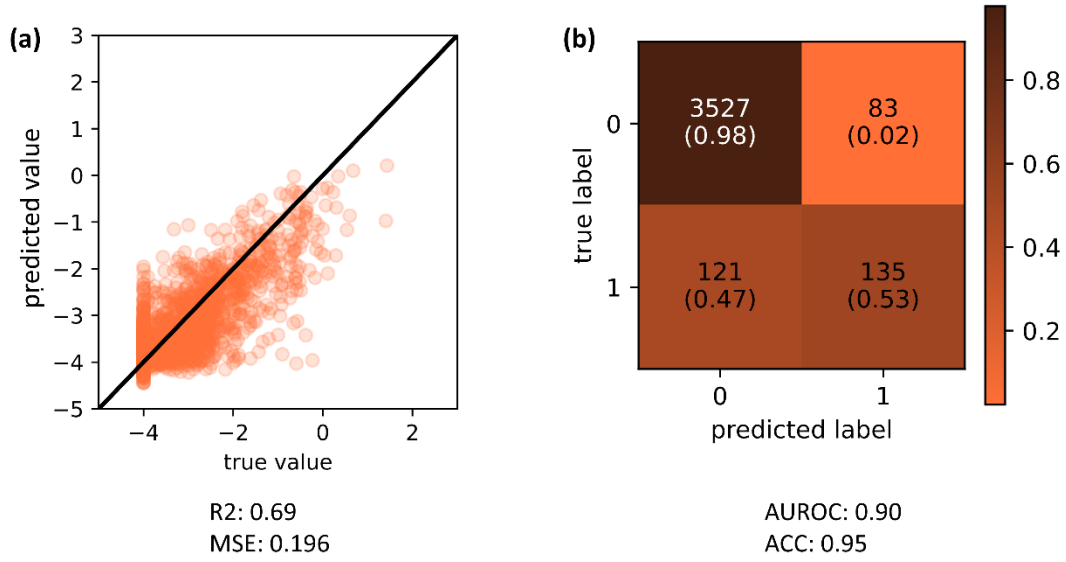

Figure 5. DeepLPI performance on the Davis dataset, (a) comparing in regression the predicted pK_d value and the true value and (b) the confusion matrix in classification

The model regression performances on the Davis dataset were also compared and summarized in the Table 3.. The result shows that the DeepLPI obtained an R^2 of 0.70, which. means the Pearson correlation between prediction and true values is better than 0.8. The mean squared error (MSE) in theDeepLPI was 0.196, which was slightly better than the DeepDTA's 0.215 and DeepCDA's 0.208, and similar to the DTITR's 0.192. These result further demonstrated the transferability of the DeepLPI model because it was first designed for classification tasks instead of regression tasks, and the model's hyperparameters were optimized based on the classification task in the BindingDB. In the Davis dataset, there existed large amounts of interactions valued 10000 nM, indicating a non-binding experiment measurement between the pair of drug and protein. The unbalanced distribution may cause the wrong estimation in the model, which decrease the R^2 value.

Table 3. Comparing regression performance of DeepLPI, DeepCDA and DeepDTA on the internal independent testing set from the Davis data

	R^2	MSE
DeepLPI	0.70	**0.196**
DeepCDA	0.74	0.208
DeepDTA	0.75	0.215
DTITR	**0.77**	**0.192**

Evaluation on COVID-19

I further applied the model trained on the BindingDB dataset directly on the Covid-19 dataset without fine-tuning. The DeepLPI outperformed DeepCDA with an AUROC of 0.610 (Table 4). The high prediction performance on the Covid-19 dataset suggested that the DeepLPI could be the candidate method to discover effective drugs for SARS-CoV-2. However, the low PPV and specificity of the DeepLPI arised from the high false positive rates and indicated the need for further upgrades in the future works.

Table 4. Comparison of DeepLPI and DeepCDA on transferring BindingDB trained model to COVID-19.

	AUROC	**Sensitivity**	**Specificity**	**PPV**	**NPV**
DeepLPI	**0.610**	0.538	0.576	**0.110**	**0.928**
DeepCDA	0.400	0.000	**1.000**	nan	0.911

Discussion

In the work, DeepLPI mode was successfully built to predict DTI in classification tasks using 1D sequence data from protein and drug molecules. First utilized are the pre-trained embedding methods called Mol2Vec and ProSE to encode the raw drug molecular SMILES strings and target protein sequences into dense vector representations. Then, the encoded dense vector representations are fed separately into head modules and ResNet-based modules to extract features, where these modules are based on 1D CNN. The extracted feature vectors are concatenated and fed into the biLSTM network, further followed by the MLP module to finally predict binary active or inactive based on Kd affinity labeled data. Three datasets of BindingDB, Davis and COVID-19 were used to evaluate the DeepLPI model, and the results demonstrate that the model has a high performance on the prediction. The model is also adapted to do the regression on the Davis dataset, and the result is compatible with current methods. For the recently published DTITR [27], which employed transformer and cross attentions for ligand-protein binding affinity prediction, it is compared with the DeepLPI model. It is found that they

used a very different data preprocessing procedure and adjusted the hyperparameters on a different database. Specifically, the DTITR dealt with drug information encoding using a dictionary that didn't include "P" for phosphorus element and treated the letters in SMILES representations case-sensitively. In this case, this model cannot be adapted to the BindingDB database and compared with DeepLPI straightforwardly on equal basis. As a temporary solution, the DTITR and DeepLPI methods are compared on the regression task and on the DTITR-specified Davis dataset. Although the results show DTITR has slightly better performance, DeepLPI's result is also compatible (DTITR AUROC 0.932, MSE 0.192, DeepLPI AUROC 0.923, MSE 0.196). Future study is guaranteed to develop a unified data preprocessing procedure applicable to DTITR and DeepLPI.

Unlike the methods to pre-define features that rely heavily on domain knowledge or to represent sequences simply using a sparse encoding approach, the DeepLPI applied pre-trained embedding models of Mol2Vec[17] and ProSE[18] to encode the raw drug SMILES string and target protein sequences, respectively. These semantic context embedding models are trained using a huge dataset to represent sequence data in dense vectors, considering the structure information of molecules and target proteins to ensure that they are highly informative and efficient for feature embeddings. The language model-based contextual embedding is assumed to be the primary reason DeepLPI outperforms DeepCDA and DeepDTA. It is admired that there exists a variety of embedding methods to encode drug compounds and protein sequences. the method used a protein language model (ProSE) for embedding, which outperformed a couple of baseline language models [28, 29]. In the future study, it is desirable to further compare the performance of ProSE and other novel protein language models. 1D CNN is used in the DeepLPI model to retain the sequential correlation, and a ResNet-based module is adopted in the DeepLPI. Traditional feed-forward CNN may lose useful information as the design grows deeper. Nevertheless, ResNet-based CNN can mitigate this drawback by developing a "shortcut connection" for the network. Consequently, data inputted into the ResNet-based CNN module can be added with the residual of the network to alleviate the loss of information. The biLSTM is employed in the DeepLPI model, which can capture long-term dependencies of the sequence and equally encode the input sequence from two sides of it. Compared to the classical LSTM, the biLSTM enables the use of the two hidden states in each LSTM memory block to preserve information from both the past and the future, which means keeping the memory of protein when processing drugs and keeping the memory of drugs when processing the proteins.

In the experiments, it is noticed that the performance of DeepLPI is not uniform on different proteins: some common biological features of those proteins might exist, such as the sequences or the spatial structures. Detailed analysis of the shared features of the proteins requires a deeper understanding of the protein-drug interaction and can potentially explain why the model behaves well on some of the proteins. Such analysis would be useful to improve the model upon generalizing the results later.

The DeepLPI model may help in speeding up COVID-19 drug research. As of today, people are still searching for an effective and safe cure for Covid-19 patients. The current widely-used combination treatment with hydroxychloroquine and azithromycin has not been proven to be satisfactory, and there are some research efforts in using computational, especially deep neural network, techniques for searching the effective repurposed drugs. The model can be useful in speeding up the drug search and potentially increasing the success rate because the training data fed into the model is not limited to the protein structural information.

Even though the model has been successfully built to predict active/inactive interaction with high accuracy, it has certain limitations. There is still room for improvement regarding prediction accuracy, especially when the model is applied to external datasets. To rigorously evaluate model prediction power, a control test set had better be significantly different from the training set in distributions. This work's "unseen" test sets only differ in molecular and protein length distributions. Meanwhile, other splitting methods exist, such as those characterized by molecular scaffold or more quantitatively protein similarity. Comparing with alternative differential data splitting approaches is out of the scope of this study, but the impact will be investigated of different data splitting approaches in future work. From a broader perspective, the study of repurposing drugs should not be limited only to the binding affinities. Researchers should also pay attention to the possibility of potential adverse effects of using the repurposed drug. This can result from new interactions between the drug and the proposed disease target, or because the drug is administered to a new population group. Sometimes the repurposed drug could interact with traditional drugs on the new disease, and adverse effects might also arise from such unexpected interactions. Deep learning methods could also be used in studies on these aspects for better safety.

Acknowledgements

The project acknowledge inspiration from Prof. Shuyun Dong at the Department of Pharmaceutics and Pharmaceutical Chemistry, University of Utah for insights on drug repurposing and drug-protein binding affinities, and for interesting discussions on lab experiments.

Appendix

In August 2022, the major progress of this project formed a manuscript which was submitted to the journal Scientific Reports. In October 2022, the manuscript was fortunately accepted for publication. During this process, various model optimizations was continually tried out but failed. Very recently, a breakthrough was made which significantly raised the model's prediction accuracy, and this report submitted to Regeneron Science Talent Search reflected the most recent breakthrough, with updated analysis and illustrations. The outdated figures in the published manuscript were reserved in this Appendix section for the reviewers' reference.

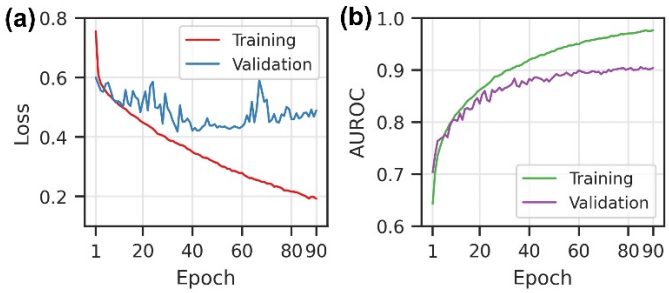

Figure A1 (Outdated DeepLPI model) The loss and AUROC score during the DeepLPI training on the BindingDB Kd dataset. (a) Loss scores for training and validation. (b) AUROC scores for training and validation

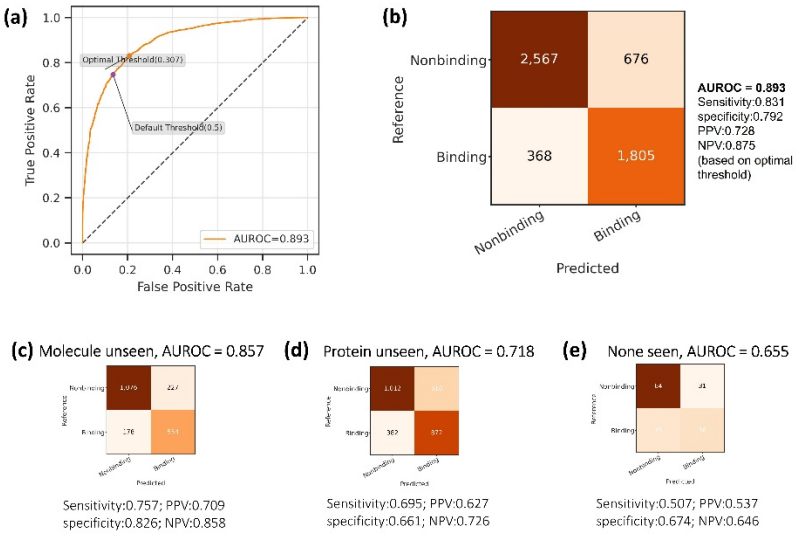

Figure A2. (Outdated DeepLPI model) The prediction performance of the final DeepLPI model on BindingDB dataset. (a) The ROC curve and the determined optimal threshold using Youden's J statistics. (b) Confusion matrix based on the optimal threshold. (c) – (e) Confusion matrix and performance metrics on the three "unseen" drug/protein testsets: (c) Molecule unseen (d) Protein unseen and (e) None seen.

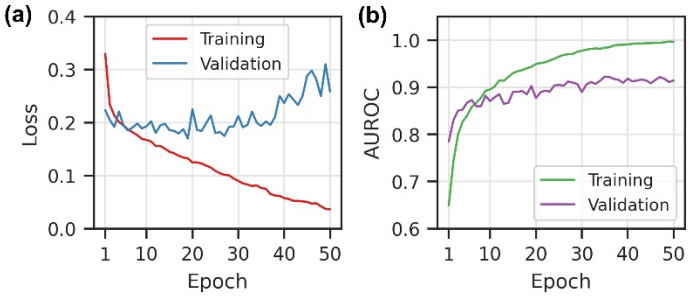

Figure A3. (Outdated DeepLPI model) The loss and AUROC score during the DeepLPI training on Davis dataset **(a)** Loss scores for training and validation. **(b)** AUROC scores for training and validation

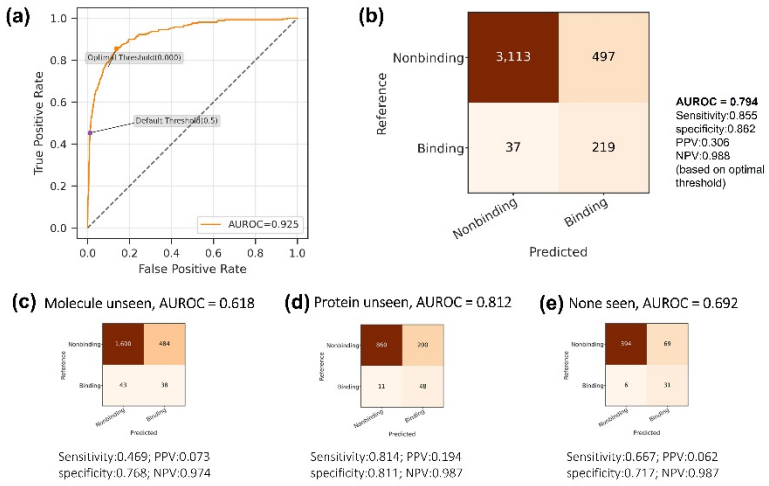

Figure A4. (Outdated DeepLPI model) The prediction performance of the final DeepLPI model Davis dataset. (a) The ROC curve and the determined optimal threshold. (b) Confusion matrix based on the optimal threshold. (c) – (e) Confusion matrix and performance metrics on the three unseen drug/protein testsets: (c) Molecule unseen (d) Protein unseen and (e) None seen.

References

[1] S. Pushpakom *et al.*, “Drug repurposing: progress, challenges and recommendations,” *Nature Reviews Drug Discovery*, vol. 18, no. 1, Jan. 2019, doi: 10.1038/nrd.2018.168.

[2] O. J. Wouters, M. McKee, and J. Luyten, "Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018," *JAMA*, vol. 323, no. 9, Mar. 2020, doi: 10.1001/jama.2020.1166.

[3] R. Mahajan and K. Gupta, “Food and drug administration′s critical path initiative and innovations in drug development paradigm: Challenges, progress, and controversies,” *Journal of Pharmacy and Bioallied Sciences*, vol. 2, no. 4, 2010, doi: 10.4103/0975-7406.72130.

[4] F. Huang *et al.*, “Identification of amitriptyline HCl, flavin adenine dinucleotide, azacitidine and calcitriol as repurposing drugs for influenza A H5N1 virus-induced lung injury,” *PLOS Pathogens*, vol. 16, no. 3, Mar. 2020, doi: 10.1371/journal.ppat.1008341.

[5] P. Sun, J. Guo, R. Winnenburg, and J. Baumbach, “Drug repurposing by integrated literature mining and drug–gene–disease triangulation,” *Drug Discovery Today*, vol. 22, no. 4, Apr. 2017, doi: 10.1016/j.drudis.2016.10.008.

[6] T. B. Kimber, Y. Chen, and A. Volkamer, “Deep Learning in Virtual Screening: Recent Applications and Developments,” *International Journal of Molecular Sciences*, vol. 22, no. 9, Apr. 2021, doi: 10.3390/ijms22094435.

[7] I. Wallach, M. Dzamba, and A. Heifets, “AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery.” *arXiv preprint* arXiv:1510.02855 (2015).

[8] S. Wang *et al.*, "SE-OnionNet: A Convolution Neural Network for Protein–Ligand Binding Affinity Prediction," *Frontiers in Genetics*, vol. 11, Feb. 2021, doi: 10.3389/fgene.2020.607824.

[9] Z. Liu *et al.*, “PDB-wide collection of binding data: current status of the PDBbind database,” *Bioinformatics*, vol. 31, no. 3, Feb. 2015, doi: 10.1093/bioinformatics/btu626.

[10] Jumper, J., Evans, R., Pritzel, A. et al. “Highly accurate protein structure prediction with AlphaFold.” *Nature* **596**, 583–589 (2021). https://doi.org/10.1038/s41586-021-03819-2.

[11] A. Bayat, “Science, medicine, and the future: Bioinformatics,” *BMJ*, vol. 324, no. 7344, (2002), doi: 10.1136/bmj.324.7344.1018.

[12] P. J. Ballester and J. B. O. Mitchell, “A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking,” *Bioinformatics*, vol. 26, no. 9, May 2010, doi: 10.1093/bioinformatics/btq112.

[13] H. Li, K.-S. Leung, M.-H. Wong, and P. Ballester, “Low-Quality Structural and Interaction Data Improves Binding Affinity Prediction via Random Forest,” *Molecules*, vol. 20, no. 6, Jun. 2015, doi: 10.3390/molecules200610947.

[14] P. -W. Hu, K. C. C. Chan and Z. -H. You, “Large-scale prediction of drug-target interactions from deep representations,” *2016 International Joint Conference on Neural Networks* (IJCNN), 2016, pp. 1236-1243, doi: 10.1109/IJCNN.2016.7727339.

[15] H. Öztürk, A. Özgür, and E. Ozkirimli, “DeepDTA: Deep drug-target binding affinity prediction,” in *Bioinformatics*, Sep. 2018, vol. 34, no. 17, pp. i821–i829. doi: 10.1093/bioinformatics/bty593.

[16] K. Abbasi, P. Razzaghi, A. Poso, M. Amanlou, J. B. Ghasemi, and A. Masoudi-Nejad, “DeepCDA: deep cross-domain compound–protein affinity prediction through LSTM and convolutional neural networks,” *Bioinformatics*, vol. 36, no. 17, Nov. 2020, doi: 10.1093/bioinformatics/btaa544.

[17] S. Jaeger, S. Fulle, and S. Turk, “Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition.” J. Chem. Inf. Model. 58, 1, 27–35 (2018) doi: 10.1021/acs.jcim.7b00616

[18] T. Bepler and B. Berger, “Learning the protein language: Evolution, structure, and function,” *Cell Systems*, vol. 12, no. 6, Jun. 2021, doi: 10.1016/j.cels.2021.05.017.

[19] T. Liu, Y. Lin, X. Wen, R. N. Jorissen, and M. K. Gilson, “BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities,” *Nucleic Acids Research*, vol. 35, Issue suppl_1, P D198–D201 (2007) doi: 10.1093/nar/gkl999.

[20] M. I. Davis *et al.*, “Comprehensive analysis of kinase inhibitor selectivity,” *Nature Biotechnology*, vol. 29, no. 11, Nov. 2011, doi: 10.1038/nbt.1990.

[21]Diamond Light Source, “Main protease structure and XChem fragment screen,” *Online data source.* https://www.diamond.ac.uk/covid-19/for-scientists/Main-protease-structure-and-XChem.html.

[22] Huang, K. et al , “Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development”, *arXiv preprints*, 2021. arXiv:2102.09548

[23] D. Rogers and M. Hahn, “Extended-Connectivity Fingerprints,” *Journal of Chemical Information and Modeling*, vol. 50, no. 5, May 2010, doi: 10.1021/ci100050t.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *2016 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *2015 IEEE International Conference on Computer Vision (ICCV),* 2015, pp. 1026-1034, doi: 10.1109/ICCV.2015.123.

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint* arXiv:1412.6980 (2014).

[27] N. R. C. Monteiro, J. L. Oliveira and J. P. Arrais, "DTITR: End-to-end drug–target binding affinity prediction with transformers," *Computers in Biology and Medicine*, vol. 147, 105772, 2022, https://doi.org/10.1016/j.compbiomed.2022.105772

[28] M. Gardner, J. Grus, M. Neumann, *et al*, "AllenNLP: A Deep Semantic Natural Language Processing Platform," *arXiv preprint* arXiv:1803.07640 (2018).

[29] Ehsaneddin Asgari, Mohammad R.K. Mofrad, "ProtVec: A Continuous Distributed Representation of Biological Sequences," *arXiv preprint* arXiv:1503.05140 (2015)